­­­­

Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Інститут комп’ютерних наук та інформаційних технологій

Кафедра програмного забезпечення



**Звіт**

Про виконання лабораторної роботи №3

**на тему:**

«МЕТОД СОРТУВАННЯ ШЕЛЛА»

З дисципліни «Алгоритми і структури даних»

**Лектор:**

доц. каф. ПЗ

Коротєєва Т.О.

**Виконав:**

ст. гр. ПЗ-26

Комарницький В.В.

**Прийняв:**

асистент каф. ПЗ

Симець І.І.

« \_\_ » \_\_\_\_\_\_\_\_ 2022 р.

∑ = \_\_\_\_\_ \_\_\_\_\_\_\_\_\_ .

Львів-2022

**Тема роботи:** Метод сортування Шелла.

**Мета роботи:** Вивчити алгоритм сортування Шелла. Здійснити програмну реалізацію алгоритму сортування Шелла. Дослідити швидкодію алгоритму сортування Шелла.

**Теоретичний матеріал**

**Алгоритм SH**

**Індивідуальне завдання**

**Варіант 14.**

Задано перелік міст. Упорядкувати за алфавітом лише ті міста, довжина назв яких не перевищує 8.

**Код програми**

**Функція для зчитування назв міст з файлу:**

void MainWindow::**read\_cities**()

{

vector1.clear();

QFile file(QFileDialog::getOpenFileName(*this*, tr("Open file"), "", tr("Text files (\*.txt)")));

*if*(!file.*open*(QIODevice::*ReadOnly*))

{

qCritical() << file.errorString();

*return*;

}

QTextStream stream(*&file*);

QString text;

*while*(!stream.atEnd())

{

text += stream.readLine();

text += "\n";

}

QStringList list = text.split('\n');

string tmp;

int j = 0;

QString inputString = "It was introduced: [ ";

*for*(int i = 0; i < list.length()-1; ++i) {

tmp = list[i].toStdString();

*if*(tmp.length() > 8)

{

*continue*;

}

vector1.push\_back(tmp);

*if*(j == 0)

{

inputString += QString::fromStdString(vector1[j]);

++j;

*continue*;

}

inputString += (", " + QString::fromStdString(vector1[j]));

++j;

}

inputString += " ]";

ui->steps\_list->setText(inputString);

}

**Функція для зчитування назв міст з клавіатури:**

void MainWindow::**enter\_city**()

{

QStringList list = (ui->city\_list->toPlainText()).split('\n');

QString tmp;

QString inputString = "It was introduced: [ ";

vector1.clear();

int j = 0;

*for*(int i = 0; i < list.length(); ++i) {

tmp = list[i];

*if*(tmp.length() > 8)

{

*continue*;

}

vector1.push\_back(tmp.toStdString());

*if*(j == 0)

{

inputString += QString::fromStdString(vector1[j]);

++j;

*continue*;

}

inputString += (", " + QString::fromStdString(vector1[j]));

++j;

}

inputString += " ]";

ui->steps\_list->setText(inputString);

}

**Сортування вектора міст за допомогою метода Шелла:**

void MainWindow::**shellSort**(vector<string>& str, int num)

{

QString steps;

int n = 0;

*for*(int gap = num / 2; gap > 0; gap /= 2)

{

*for*(int i = gap; i < num; ++i)

{

*for*(int j = i - gap; j >= 0 && str[j] > str[j+gap]; j -= gap)

{

string tmp = str[j];

str[j] = str[j + gap];

str[j + gap] = tmp;

steps += ("Steps" + QString::number(n) + "\n[ ");

*for*(int k = 0; k < num; ++k)

{

*if*(k == 0)

{

steps += QString::fromStdString(str[k]);

*continue*;

}

steps += (", " + QString::fromStdString(str[k]));

}

steps += " ]\n";

++n;

}

}

}

ui->steps\_list->setText(steps);

}

**Протокол виконання програми**

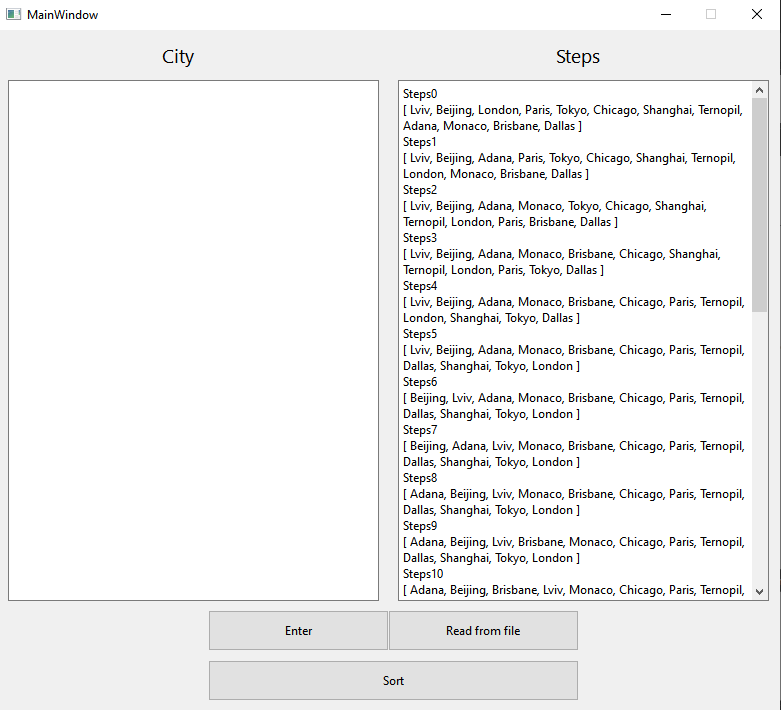


Рис.1. Сортування вектора, елементи якого введені з файлу



Рис.2. Файл з містами

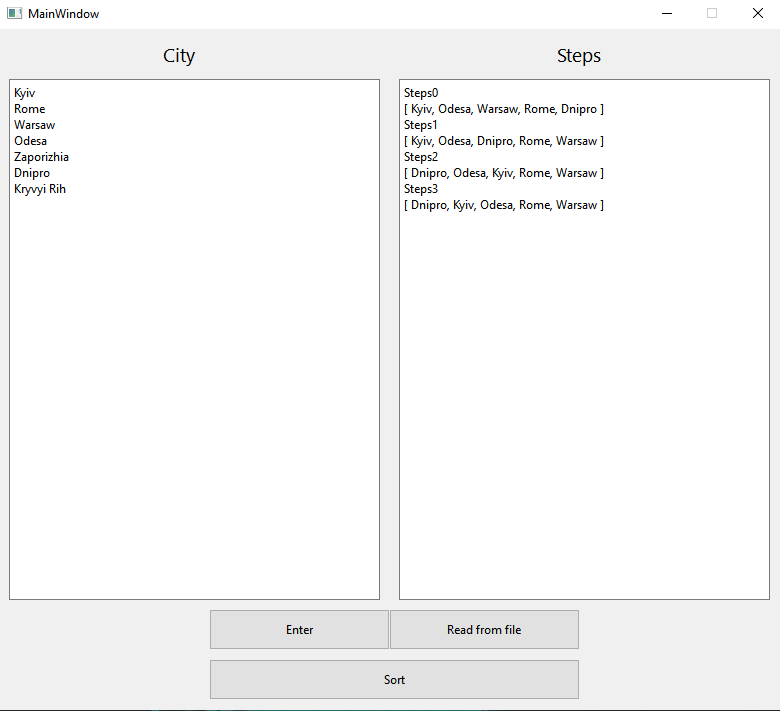


Рис.3. Сортування вектора, елементи якого введені з клавіатури

**Висновки**

Під час виконання лабораторної роботи №3 я ознайомився з алгоритмом Шелла та реалізував його в програмі з графічним інтерфейсом.